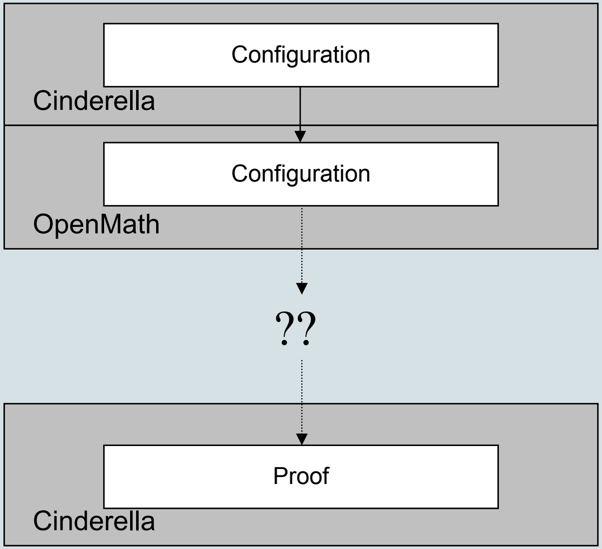
OpenMath, Bracket Proofs, and Cinderella

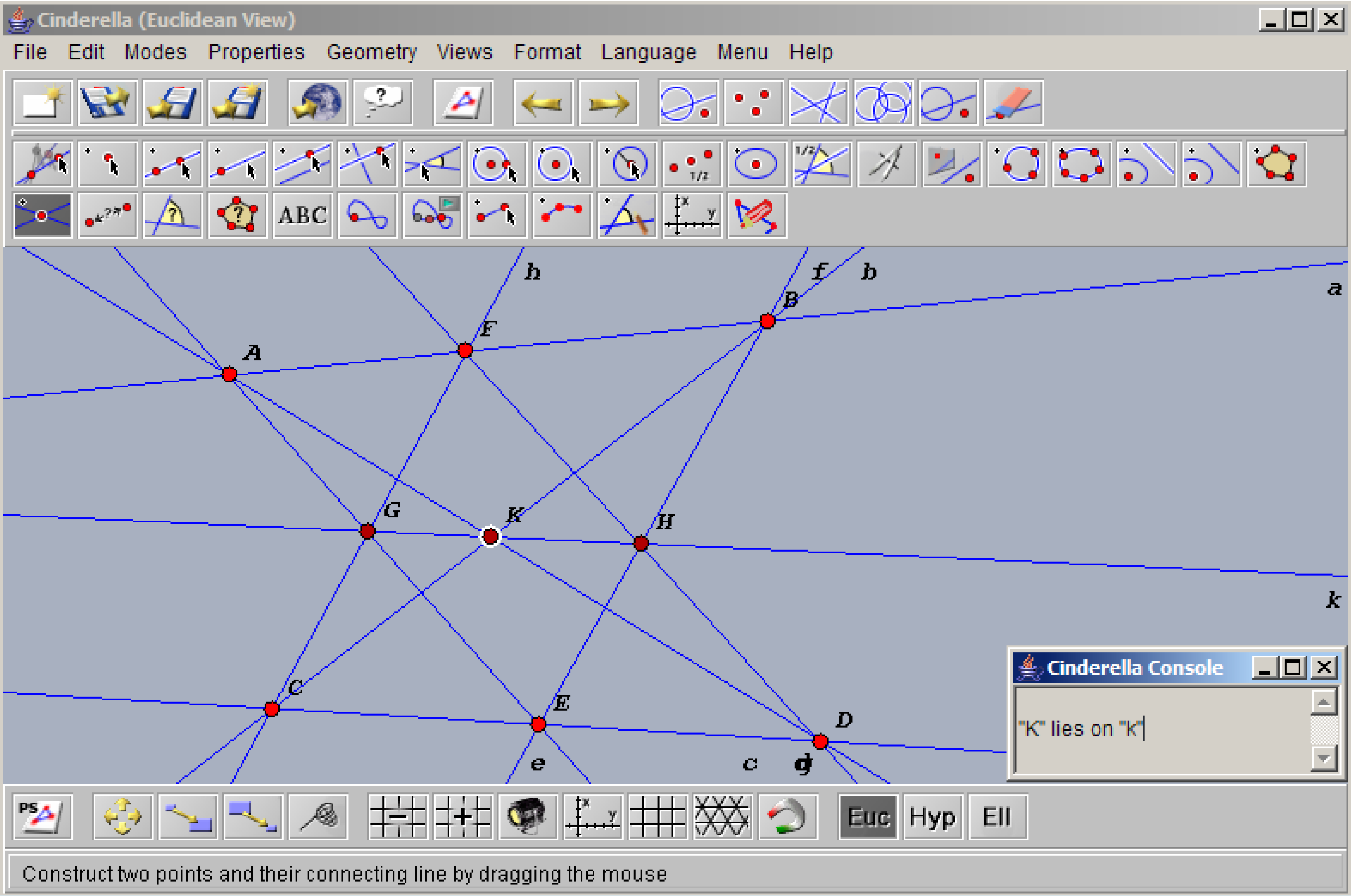
Dan Roozemond

26 March 2004

Contents

Introduction - 1




Cinderella (Euclidean View)
File Edit Modes Properties Geometry Views Format Language Menu Help
ABC
h f b a
A F B
G K H k
C E D
e c g
Cinderella Console
"K" lies on "k"
Euc Hyp Ell
Construct two points and their connecting line by dragging the mouse

Cinderella: Keywords

- Interactive Geometry Program,
- Homogeneous Coordinates,
- Euclidian / Hyperbolic / Spherical View,
- Randomized prover.

Cinderella: Construction

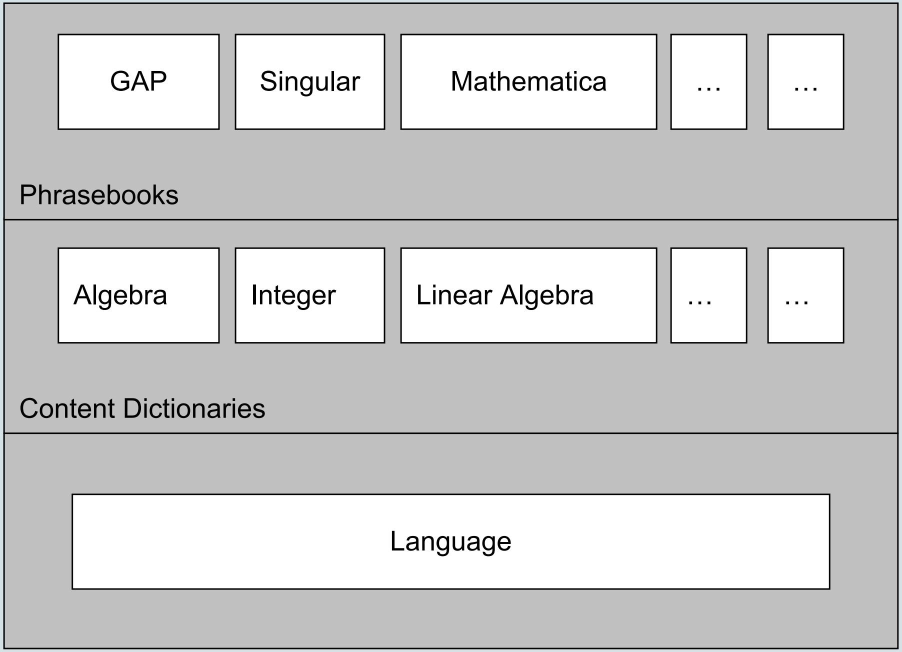
- `A = FreePoint`
- `B = FreePoint`
- `a = Join(A,B)`
- `C = PointOnLine(a)`

Cinderella: Construction

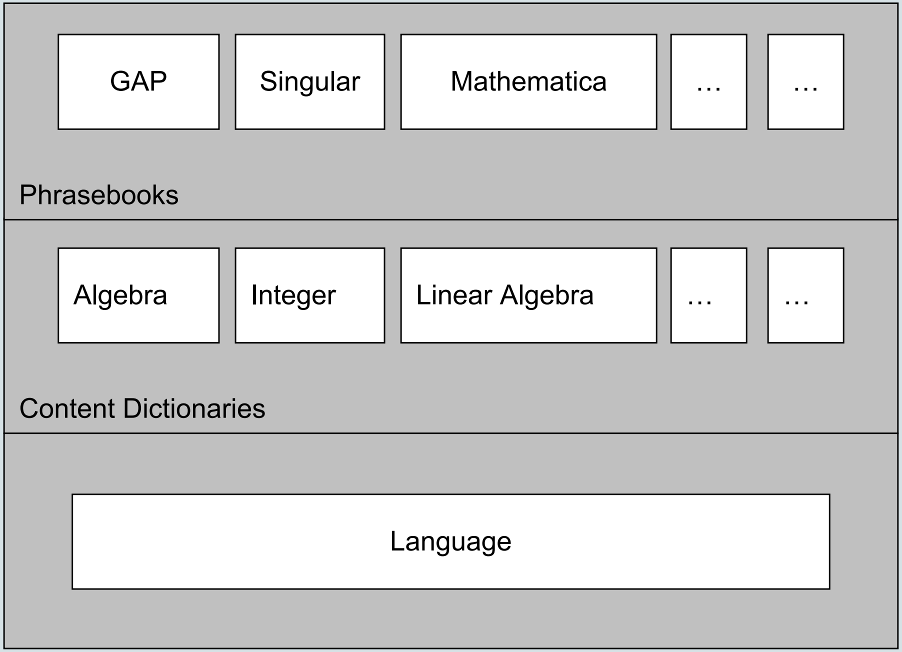
- `A = FreePoint`
- `B = FreePoint`
- `a = Join(A,B)`
- `C = PointOnLine(a)`

Easy translation to OpenMath!

OpenMath: Main

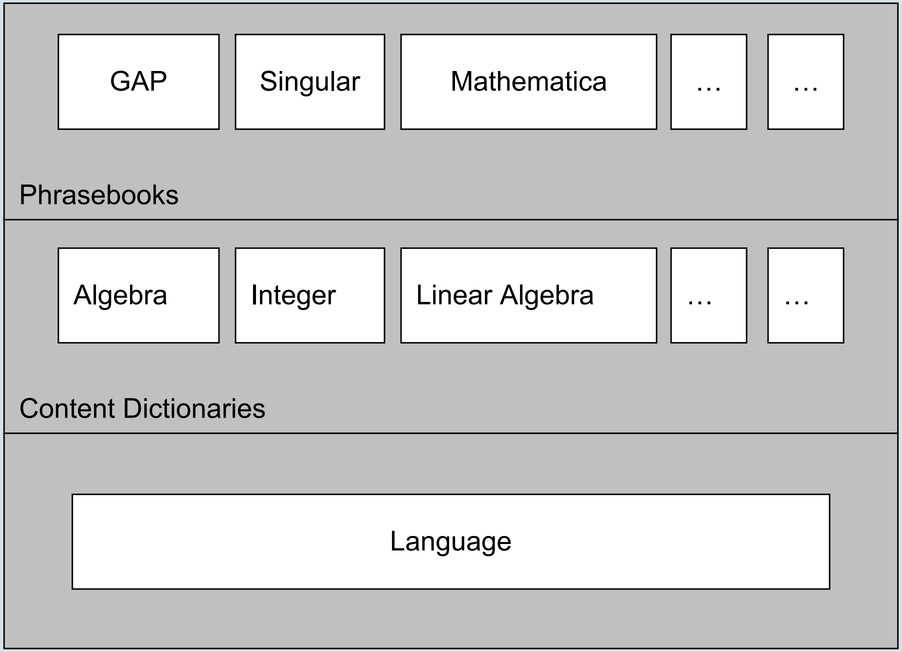


OpenMath: Main



← Planar Geometry

OpenMath: Main



← Cinderella

← Planar Geometry

OpenMath: The Language

"Place a point A and a point B in the plane. Define a to be the line through A and B"

OpenMath: The Language

- `A = FreePoint`
- `B = FreePoint`
- `a = Join(A,B)`

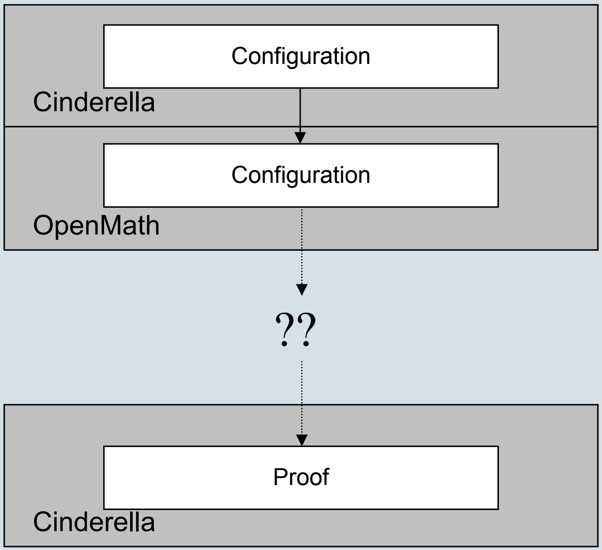
OpenMath: The Language

```
configuration (
    point(A),
    point(B),
    line(a,
        incident(a,A),
        incident(a,B)
    )
)
```

OpenMath: The Language

```
plangeo1.configuration (
    plangeo1.point(A),
    plangeo1.point(B),
    plangeo1.line(a,
        plangeo1.incident(a,A),
        plangeo1.incident(a,B)
    )
)
```

OpenMath: Conclusion



Brackets: Introduction

- Planar homogeneous coordinates,
- Assertions invariant under projective transformations,
- Calculate with *brackets*:

$$[ABC] := \begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ z_A & z_B & z_C \end{vmatrix}$$

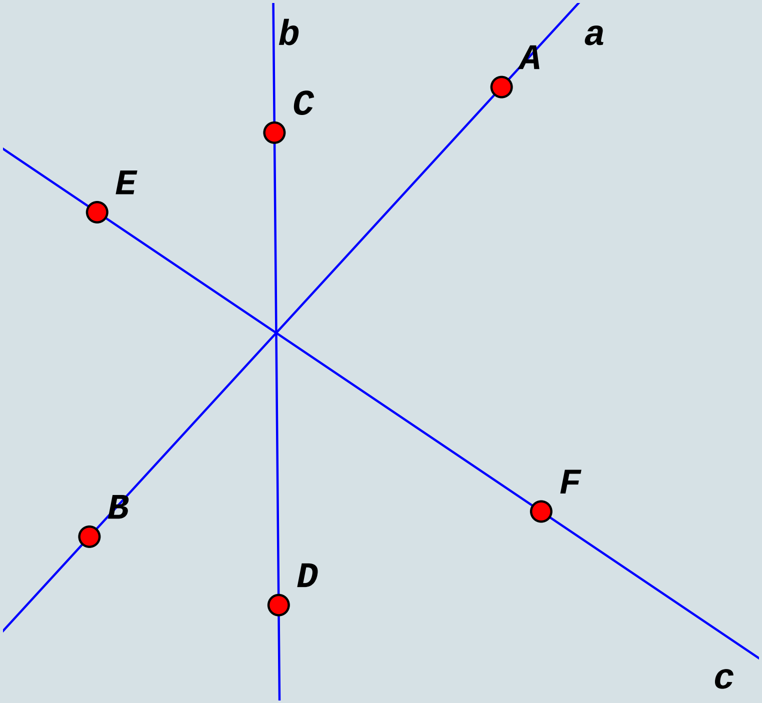
- For example: A, B, and C collinear $\Leftrightarrow [ABC] = 0$.

Brackets: Collinearity

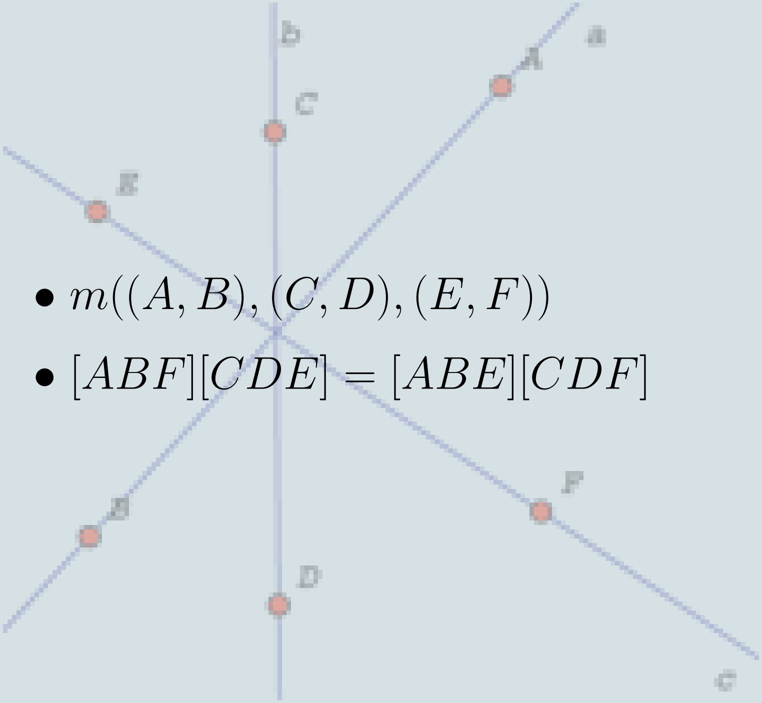
- A, B, and C collinear: $h(A, B, C)$,
- Translate to brackets:

$$[ABC] = 0 \Leftrightarrow [ABD][ACE] = [ABE][ACD].$$

Brackets: Concurrency

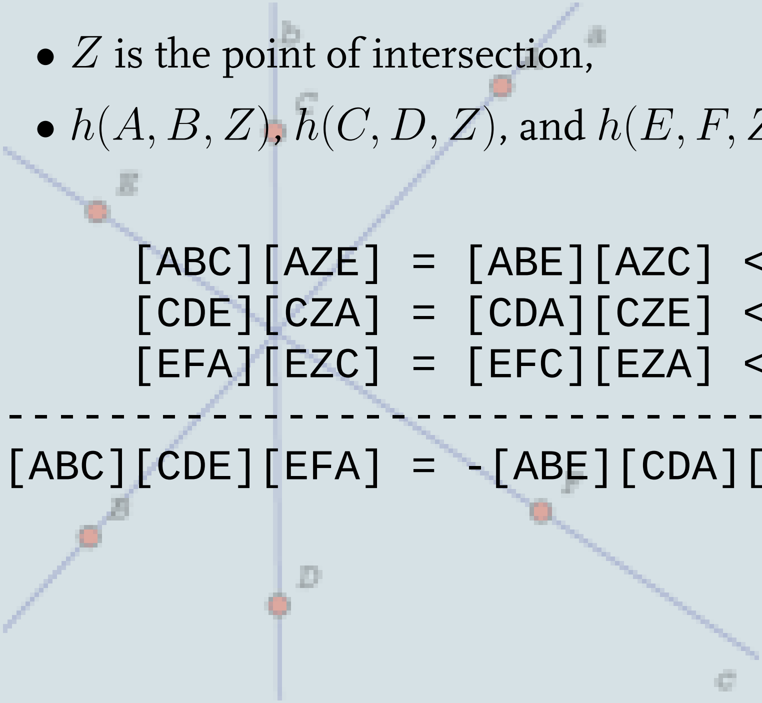


Brackets: Concurrency



- $m((A,B),(C,D),(E,F))$
- $[ABF][CDE]=[ABE][CDF]$

Brackets: Concurrency (Proof)



- Z is the point of intersection,
- $h(A, B, Z)$, $h(C, D, Z)$, and $h(E, F, Z)$.

```
    [ABC][AZE] = [ABE][AZC] <== h(A,B,Z)
    [CDE][CZA] = [CDA][CZE] <== h(C,D,Z)
    [EFA][EZC] = [EFC][EZA] <== h(E,F,Z)
-----------------------------------------
[ABC][CDE][EFA] = -[ABE][CDA][EFC]
```

Brackets: Concurrency (Proof)

- Z is the point of intersection,
- $h(A, B, Z)$, $h(C, D, Z)$, and $h(E, F, Z)$.

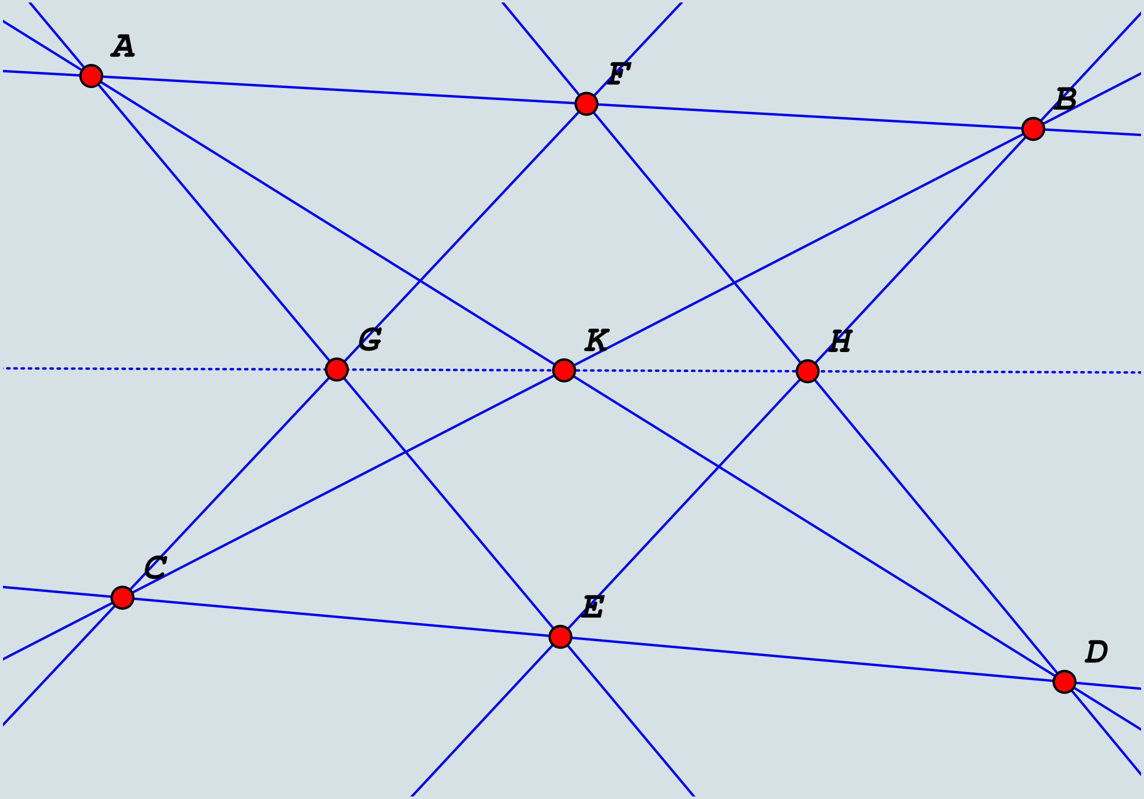
```
    [ABC][AZE] = [ABE][AZC] <== h(A,B,Z)
    [CDE][CZA] = [CDA][CZE] <== h(C,D,Z)
    [EFA][EZC] = [EFC][EZA] <== h(E,F,Z)
-----------------------------------------
[ABC][CDE][EFA] = -[ABE][CDA][EFC]
[ABF][CDA][FEC] = -[ABC][CDF][FEA]
-----------------------------------------
    [ABF][CDE] = [ABE][CDF]
```

Brackets: How to Prove

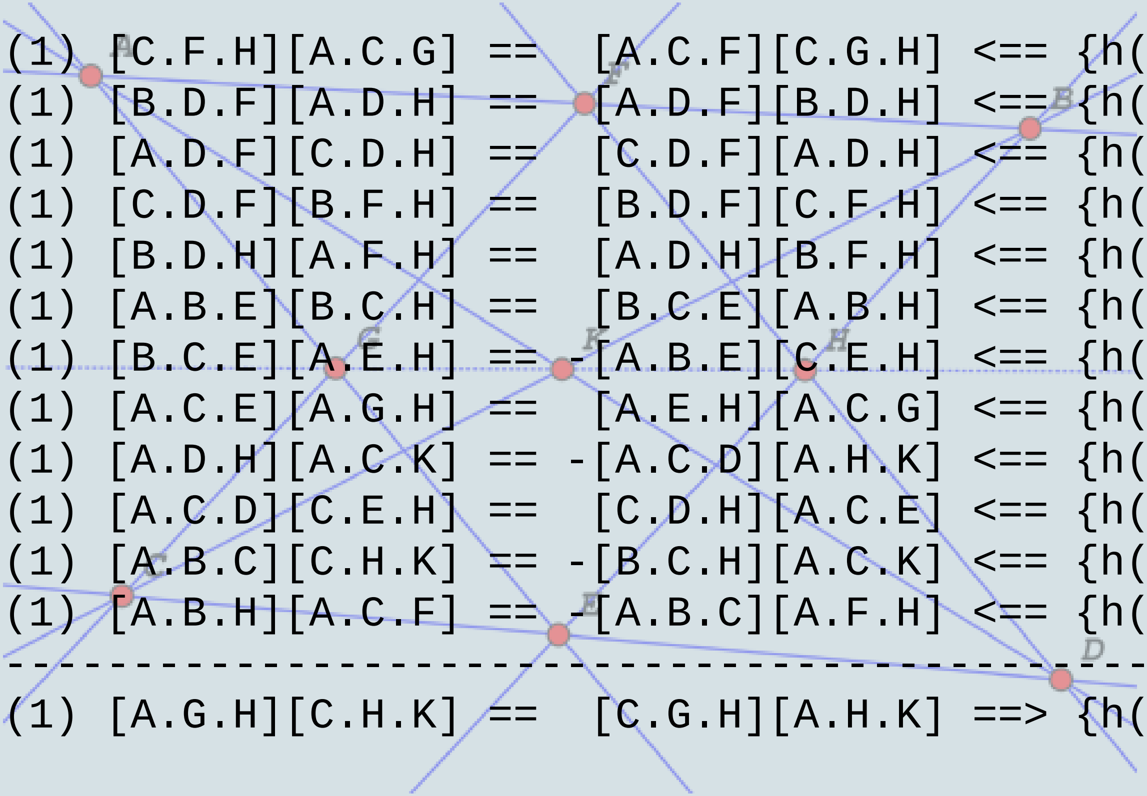
$$\begin{array}{llcl} \text{Configuration:} & [.1.][.2.] & = & [.3.][.4.] \\ & [\ldots][\ldots] & = & [\ldots][\ldots] \\ & \vdots & \vdots & \vdots \\ & [\ldots][\ldots] & = & [\ldots][\ldots] \\ \\ \text{Thesis:} & [\ldots][\ldots] & = & [\ldots][\ldots] \end{array}$$

- *Might* be solved by a system of *linear* equations....

Pappos' Theorem



Pappos' Theorem




```
(1) [C.F.H][A.C.G] ==  [A.C.F][C.G.H] <== {h(C, F, G)}
(1) [B.D.F][A.D.H] ==  [A.D.F][B.D.H] <== {h(D, F, H)}
(1) [A.D.F][C.D.H] ==  [C.D.F][A.D.H] <== {h(D, F, H)}
(1) [C.D.F][B.F.H] ==  [B.D.F][C.F.H] <== {h(D, F, H)}
(1) [B.D.H][A.F.H] ==  [A.D.H][B.F.H] <== {h(D, F, H)}
(1) [A.B.E][B.C.H] ==  [B.C.E][A.B.H] <== {h(B, E, H)}
(1) [B.C.E][A.E.H] == -[A.B.E][C.E.H] <== {h(B, E, H)}
(1) [A.C.E][A.G.H] ==  [A.E.H][A.C.G] <== {h(A, E, G)}
(1) [A.D.H][A.C.K] == -[A.C.D][A.H.K] <== {h(A, D, K)}
(1) [A.C.D][C.E.H] ==  [C.D.H][A.C.E] <== {h(C, D, E)}
(1) [A.B.C][C.H.K] == -[B.C.H][A.C.K] <== {h(B, C, K)}
(1) [A.B.H][A.C.F] == -[A.B.C][A.F.H] <== {h(A, B, F)}
-------------------------------------------------------
(1) [A.G.H][C.H.K] ==  [C.G.H][A.H.K] ==> {h(G, H, K)}
```

Brackets: Conics

Encode other projective assertions:

- $c(A, B, C, D, E, F)$:

$$[ACE][BDE][ABF][CDF] = [ABE][CDE][ACF][BDF].$$

Brackets

Advantages:

- **Quick**,

Brackets

Advantages:

- **Quick**,
- **Quick**,

Brackets

Advantages:

- **Quick**,
- **Quick**,
- Implicit non-degeneracy conditions,

Brackets

Advantages:

- **Quick**,
- **Quick**,
- Implicit non-degeneracy conditions,
- Easy to check.

Brackets

Advantages:

- **Quick**,
- **Quick**,
- Implicit non-degeneracy conditions,
- Easy to check.

Disadvantages:

- Not common knowledge,

Brackets

Advantages:

- **Quick**,
- **Quick**,
- Implicit non-degeneracy conditions,
- Easy to check.

Disadvantages:

- Not common knowledge,
- Not able to proof a theorem false,

Brackets

Advantages:

- **Quick**,
- **Quick**,
- Implicit non-degeneracy conditions,
- Easy to check.

Disadvantages:

- Not common knowledge,
- Not able to proof a theorem false,
- Implicit non-degeneracy conditions,

Brackets

Advantages:

- **Quick**,
- **Quick**,
- Implicit non-degeneracy conditions,
- Easy to check.

Disadvantages:

- Not common knowledge,
- Not able to proof a theorem false,
- Implicit non-degeneracy conditions,
- Only projective geometry.

Brackets: Non-Projective Geometry

Introduce ‘complex numbers’

$$I := \begin{pmatrix} 1 \\ i \\ 0 \end{pmatrix} \text{ and } J := \begin{pmatrix} 1 \\ -i \\ 0 \end{pmatrix},$$

Brackets: Non-Projective Geometry

Introduce ‘complex numbers’

$$I := \begin{pmatrix} 1 \\ i \\ 0 \end{pmatrix} \text{ and } J := \begin{pmatrix} 1 \\ -i \\ 0 \end{pmatrix},$$

For a point A:

$$A = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \text{ define } z_A := x + iy,$$

Brackets: Non-Projective Geometry

Introduce 'complex numbers'

$$I := \begin{pmatrix} 1 \\ i \\ 0 \end{pmatrix} \text{ and } J := \begin{pmatrix} 1 \\ -i \\ 0 \end{pmatrix},$$

For a point A:

$$A = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \text{ define } z_A := x + iy,$$

then we have

$$[ABI] = z_A - z_B \text{ and } [ABJ] = \overline{z_A - z_B}.$$

Brackets: Non-Projective Geometry

Sometimes, we can encode circles, parallel lines and perpendicular lines.

- $ci(A, B, C, D)$:

$$[ACI][BDI][ABJ][CDJ] = [ABI][CDI][ACJ][BDJ],$$

- $par((A, B), (C, D))$:

$$m((A, B), (C, D), (I, J)),$$

- $perp((A, B), (A, C))$:

$$[ABI][ACJ] = -[ABJ][ACI].$$

Brackets: Non-Projective Geometry

Sometimes, we can encode circles, parallel lines and perpendicular lines.

- $ci(A, B, C, D)$:

$$[ACI][BDI][ABJ][CDJ] = [ABI][CDI][ACJ][BDJ],$$

$$[ACE][BDE][ABF][CDF] = [ABE][CDE][ACF][BDF],$$

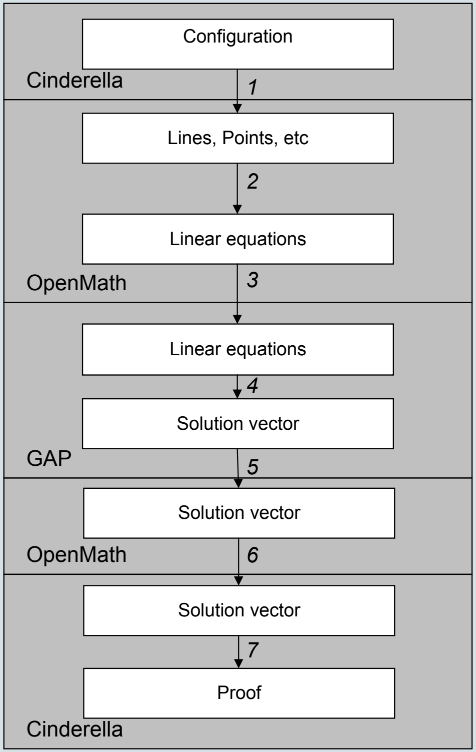
- $par((A, B), (C, D))$:

$$m((A, B), (C, D), (I, J)),$$

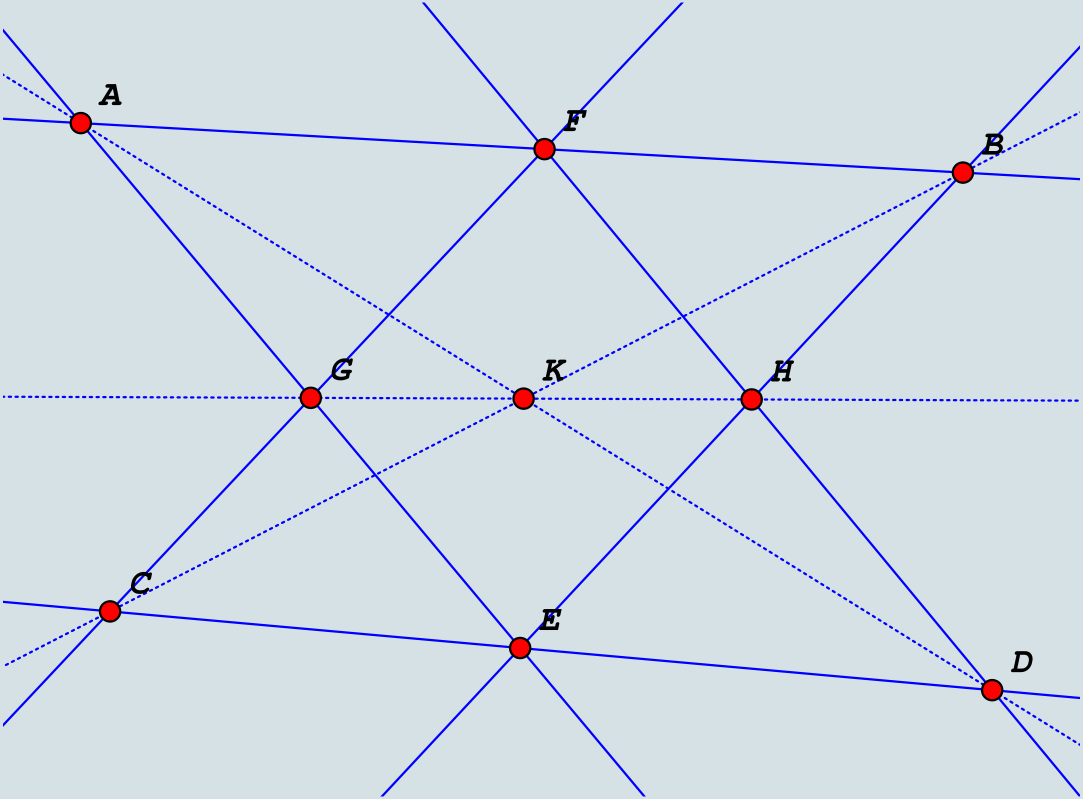
- $perp((A, B), (A, C))$:

$$[ABI][ACJ] = -[ABJ][ACI].$$

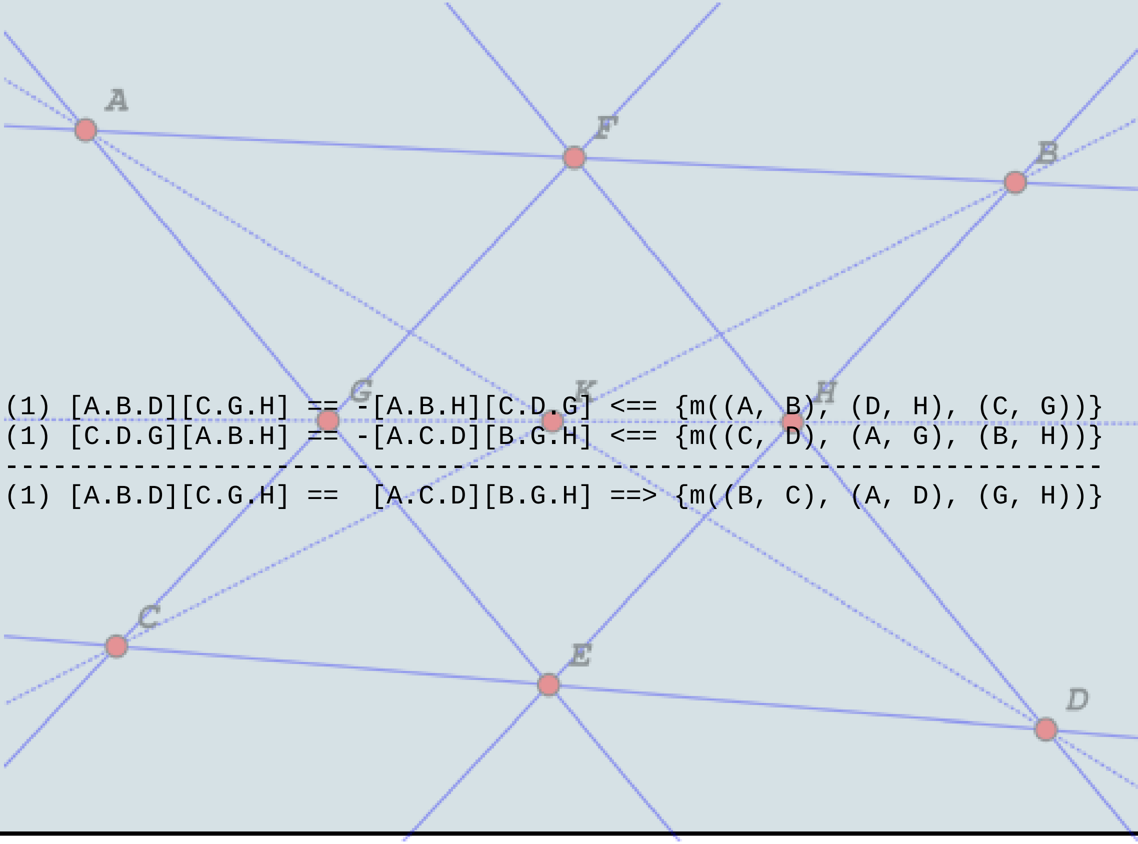
The Implementation



Examples - 1

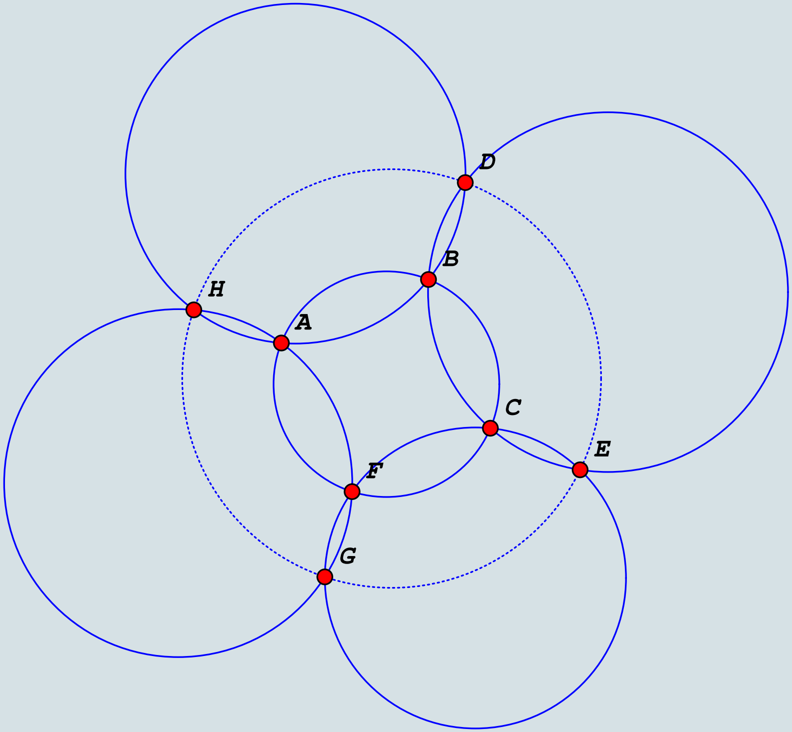


Examples - 1



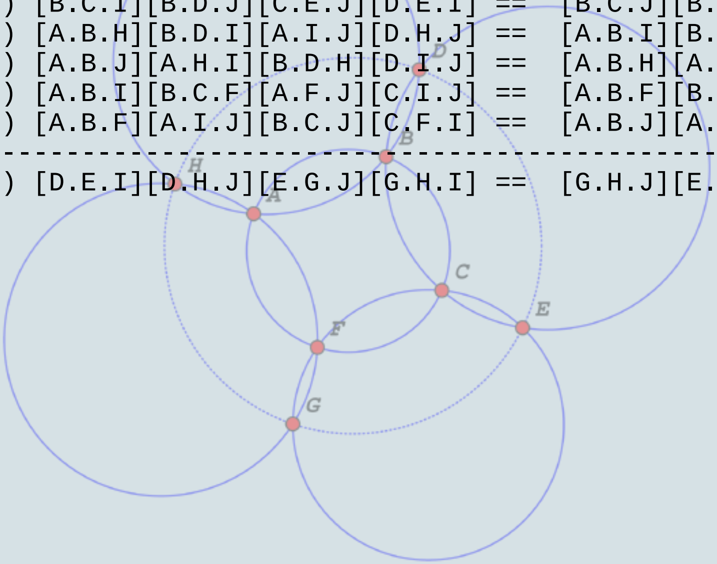

```
(1) [A.B.D][C.G.H] == -[A.B.H][C.D.G] <== {m((A, B), (D, H), (C, G))}
(1) [C.D.G][A.B.H] == -[A.C.D][B.G.H] <== {m((C, D), (A, G), (B, H))}
-------------------------------------------------------------------
(1) [A.B.D][C.G.H] ==  [A.C.D][B.G.H] ==> {m((B, C), (A, D), (G, H))}
```

Examples - 2

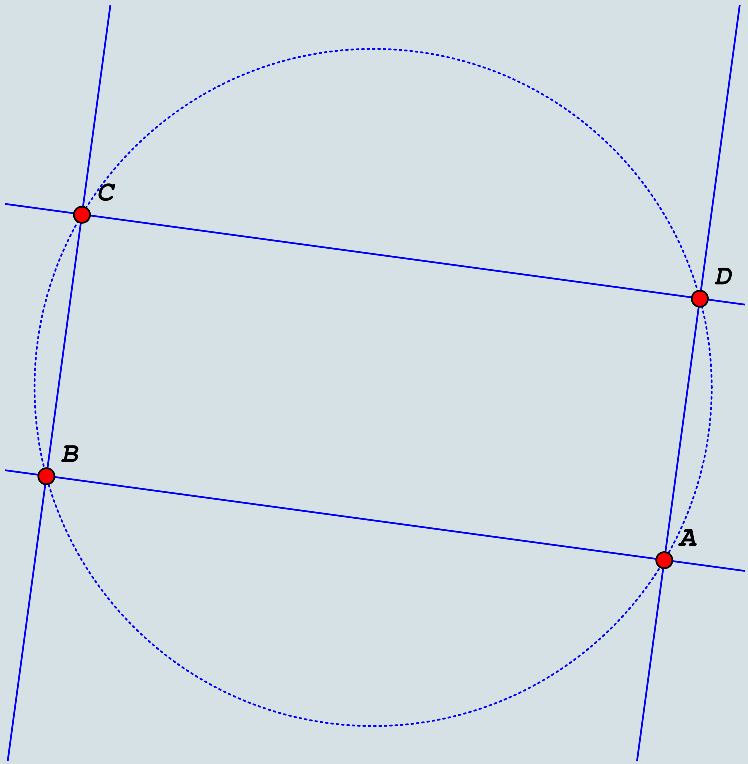


Examples - 2

```
(1) [A.F.I][F.G.H][A.H.J][G.I.J] ==  [A.F.H][F.G.I][A.I.J][G.H.J] <== {ci(A, F, G, H)}
(1) [A.F.H][A.I.J][F.G.J][G.H.I] ==  [A.F.J][A.H.I][F.G.H][G.I.J] <== {ci(A, F, G, H)}
(1) [C.E.I][C.F.J][E.G.J][F.G.I] ==  [C.E.J][C.F.I][E.G.I][F.G.J] <== {ci(C, E, F, G)}
(1) [B.C.I][B.D.J][C.E.J][D.E.I] ==  [B.C.J][B.D.I][C.E.I][D.E.J] <== {ci(B, C, D, E)}
(1) [A.B.H][B.D.I][A.I.J][D.H.J] ==  [A.B.I][B.D.H][A.H.J][D.I.J] <== {ci(A, B, D, H)}
(1) [A.B.J][A.H.I][B.D.H][D.I.J] ==  [A.B.H][A.I.J][B.D.J][D.H.I] <== {ci(A, B, D, H)}
(1) [A.B.I][B.C.F][A.F.J][C.I.J] ==  [A.B.F][B.C.I][A.I.J][C.F.J] <== {ci(A, B, C, F)}
(1) [A.B.F][A.I.J][B.C.J][C.F.I] ==  [A.B.J][A.F.I][B.C.F][C.I.J] <== {ci(A, B, C, F)}
---------------------------------------------------------------------------------------
(1) [D.E.I][D.H.J][E.G.J][G.H.I] ==  [G.H.J][E.G.I][D.H.I][D.E.J] <== {ci(D, E, G, H)}
```

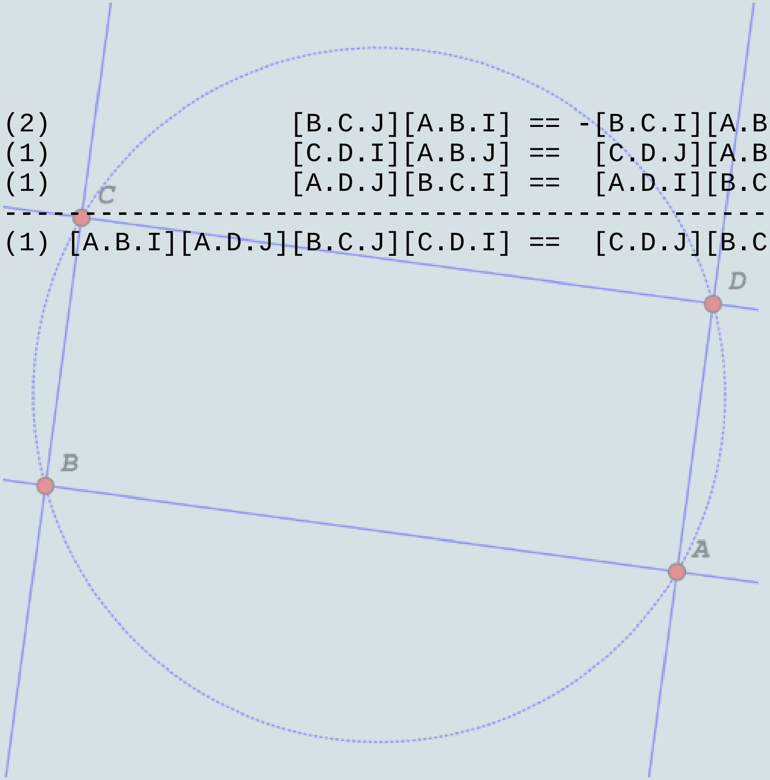


Examples - 3



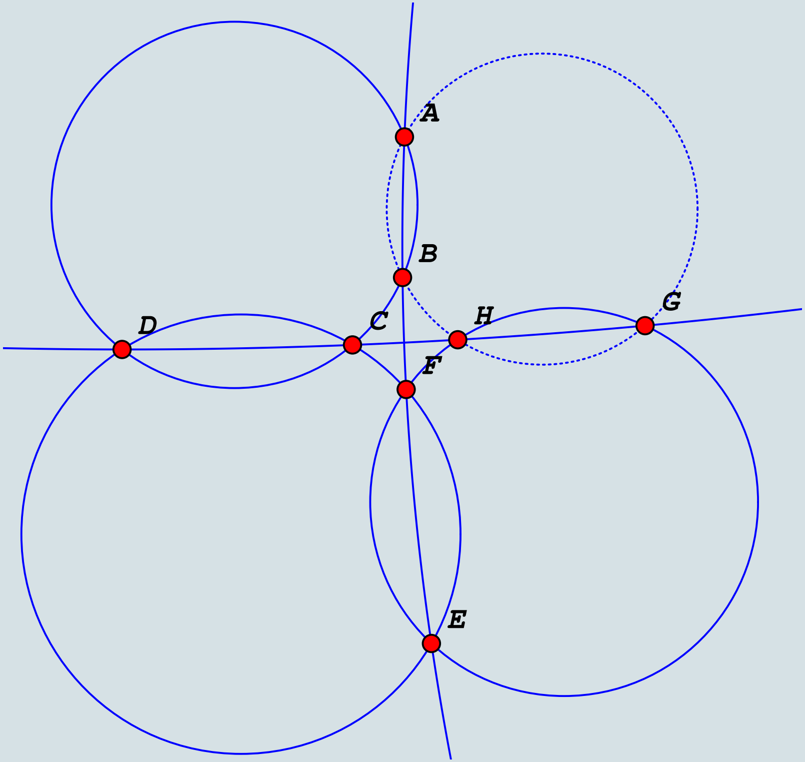
Examples - 3

```
(2)              [B.C.J][A.B.I] == -[B.C.J][A.B.J]              <== {perp((B, C), (B, A)
(1)              [C.D.I][A.B.J] ==  [C.D.J][A.B.I]              <== {par((C, D), (A, B))
(1)              [A.D.J][B.C.I] ==  [A.D.I][B.C.J]              <== {par((A, D), (B, C))
-------------------------------------------------------------------------------------------
(1) [A.B.I][A.D.J][B.C.J][C.D.I] ==  [C.D.J][B.C.I][A.D.I][A.B.J] ==> {ci(A, B, C, D)}
```



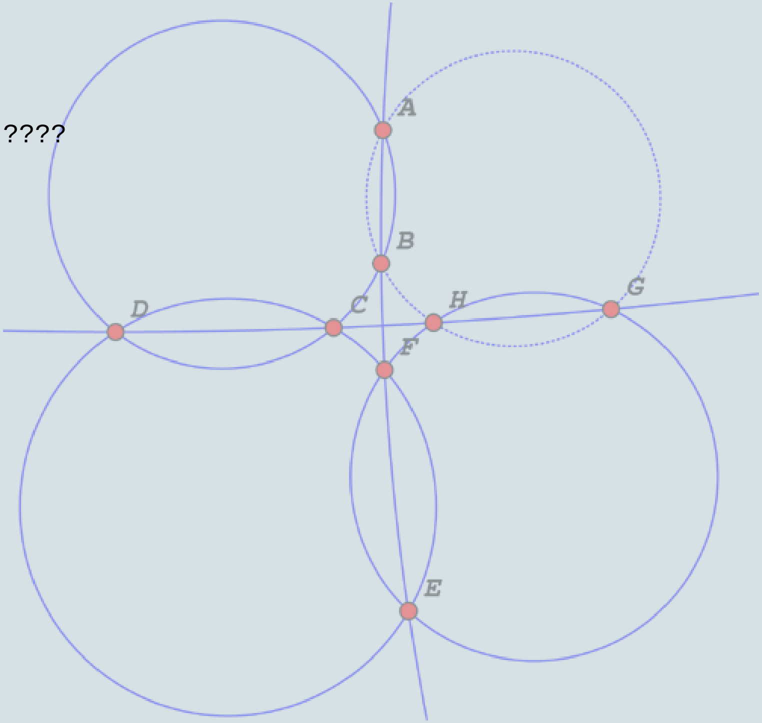
Demo

Examples - 4



Examples - 4

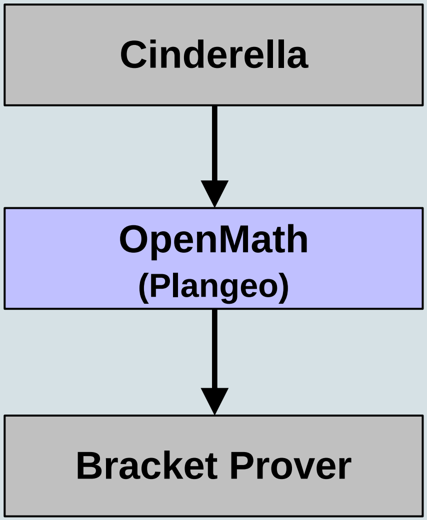
????



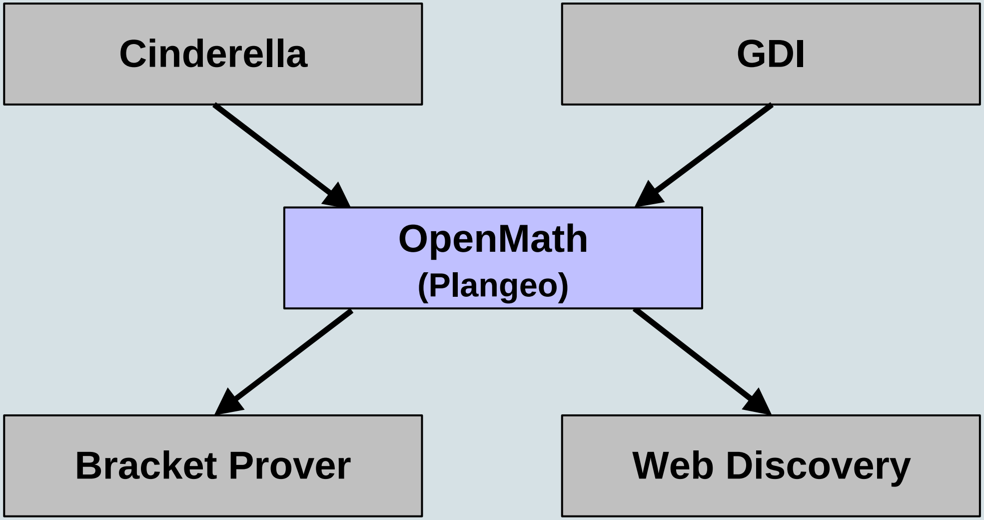
Conclusion

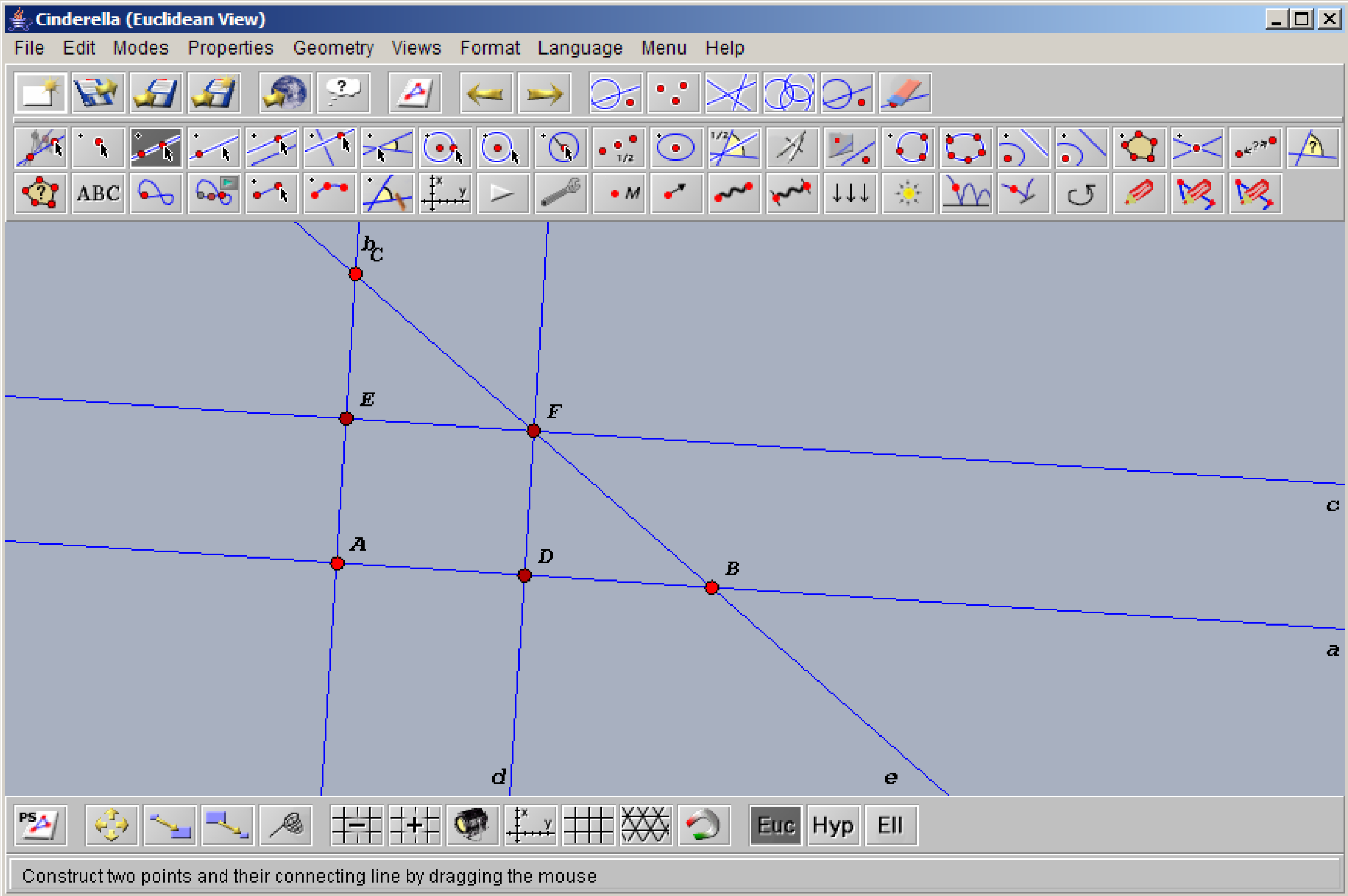
- Bracket Proofs
- OpenMath
- Future research

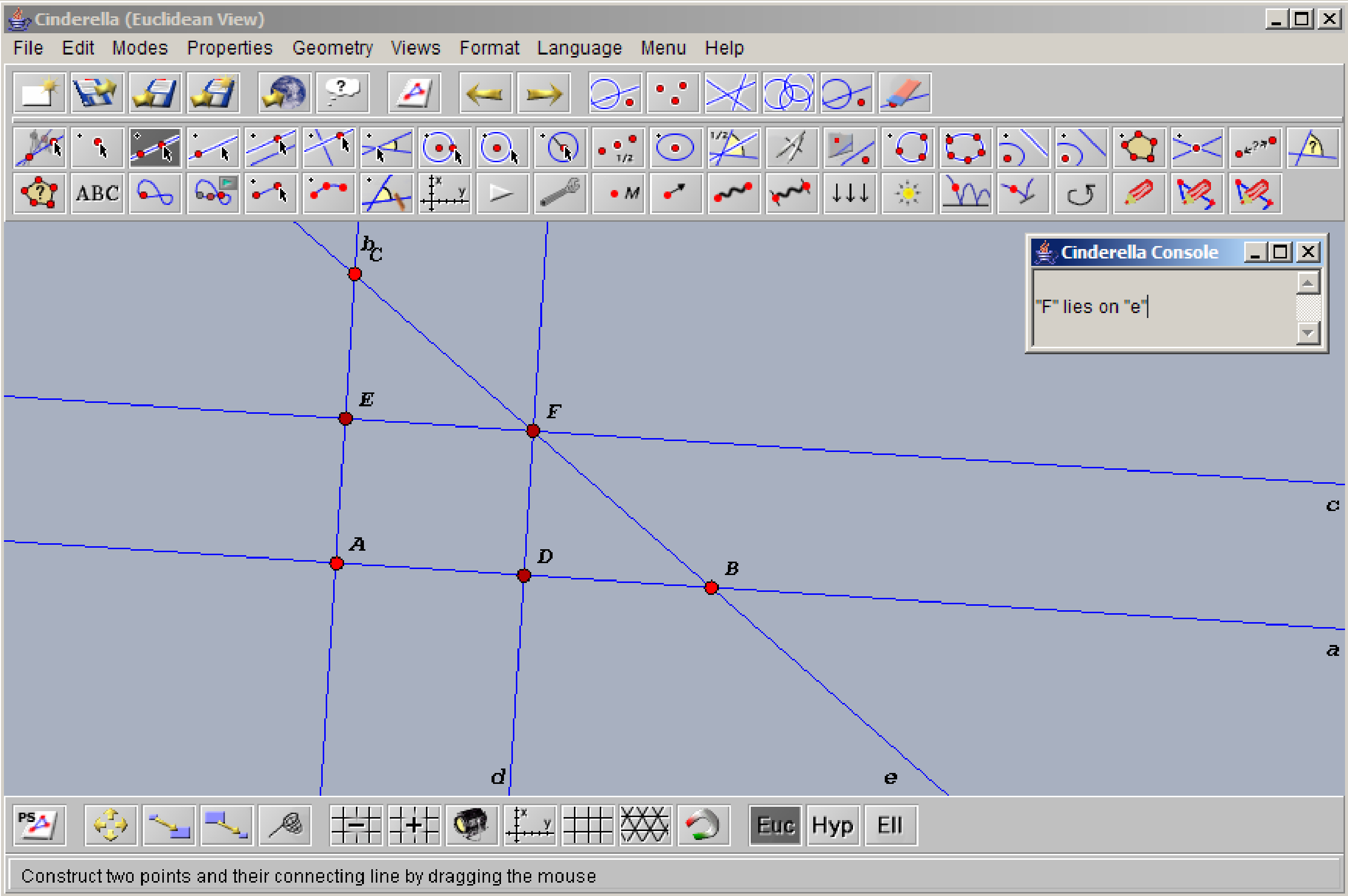
Future Research: GDI

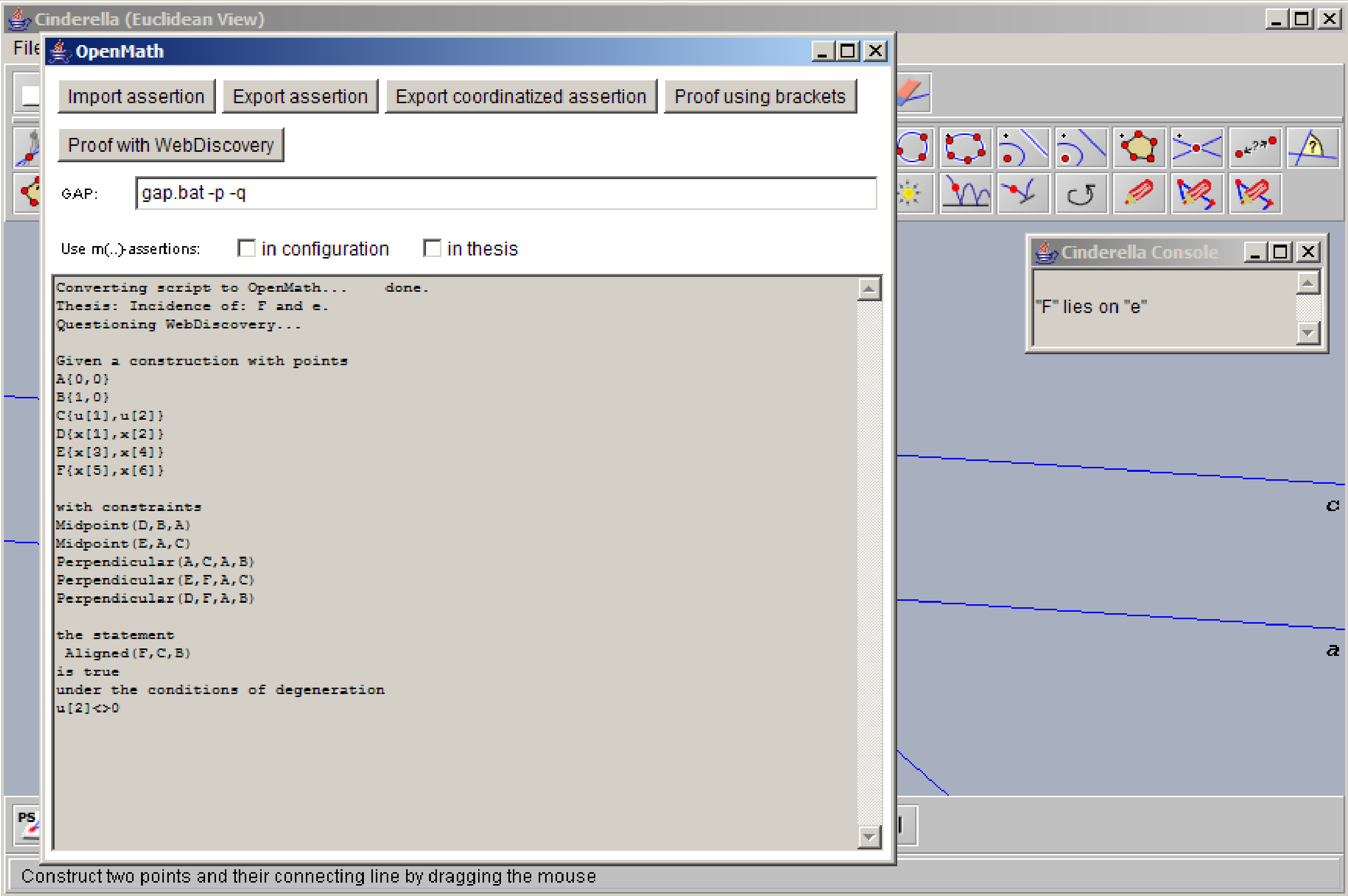


Future Research: GDI




Cinderella (Euclidean View)
File Edit Modes Properties Geometry Views Format Language Menu Help
ABC
b
C
E
F
c
A
D
B
a
d
e
Euc Hyp Ell
Construct two points and their connecting line by dragging the mouse


Cinderella (Euclidean View)
File Edit Modes Properties Geometry Views Format Language Menu Help
Cinderella Console
"F" lies on "e"
b
c
E
F
c
A
D
B
a
d
e
Euc Hyp Ell
Construct two points and their connecting line by dragging the mouse


Cinderella (Euclidean View)
File
OpenMath
Import assertion
Export assertion
Export coordinatized assertion
Proof using brackets
Proof with WebDiscovery
GAP:
gap.bat -p -q
Use m(..)-assertions:
in configuration
in thesis
Converting script to OpenMath... done.
Thesis: Incidence of: F and e.
Questioning WebDiscovery...
Given a construction with points
A{0,0}
B{1,0}
C{u[1],u[2]}
D{x[1],x[2]}
E{x[3],x[4]}
F{x[5],x[6]}
with constraints
Midpoint(D,B,A)
Midpoint(E,A,C)
Perpendicular(A,C,A,B)
Perpendicular(E,F,A,C)
Perpendicular(D,F,A,B)
the statement
Aligned(F,C,B)
is true
under the conditions of degeneration
u[2]<>0
Cinderella Console
"F" lies on "e"
c
a
Construct two points and their connecting line by dragging the mouse

Questions?